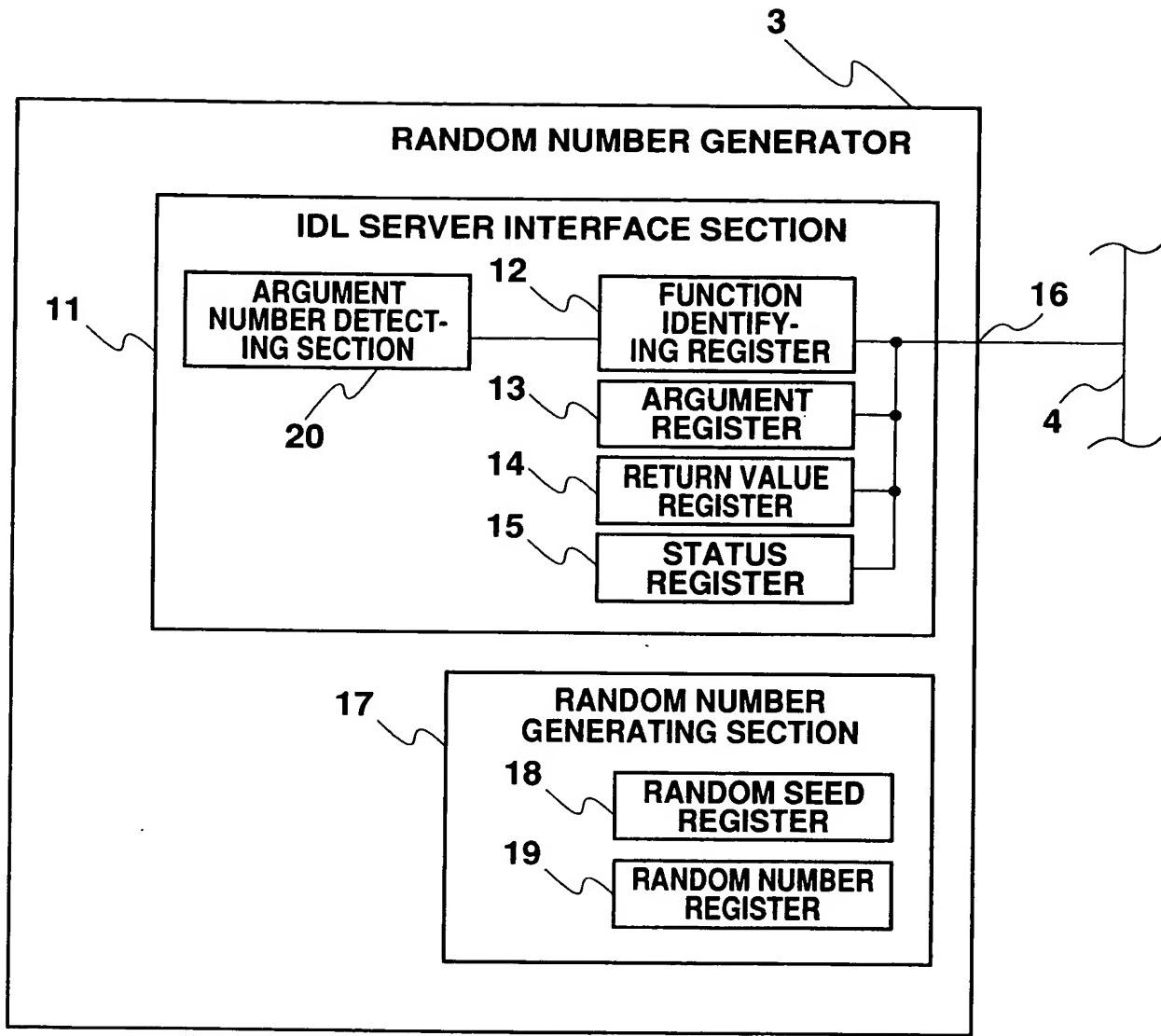


**Fig. 1**



**Fig. 2**

```
interface randomGenerator{
    void setSeed( in double seed );
    double getRandom();
};
```

**Fig. 3**

```
/* part common to IDL interface */
/* status register value */
#define Executing    1
#define Finished     2
#define Requesting   3

/* function for accessing register */
void    putFunctionID( FID );
void    putDoubleArg( double );
double  getDoubleReturn();
int    getStatus();

/* randomGenerator unique part */
/* definition of function identification value */
#define FID_setSeed    1
#define FID_getRandom  2

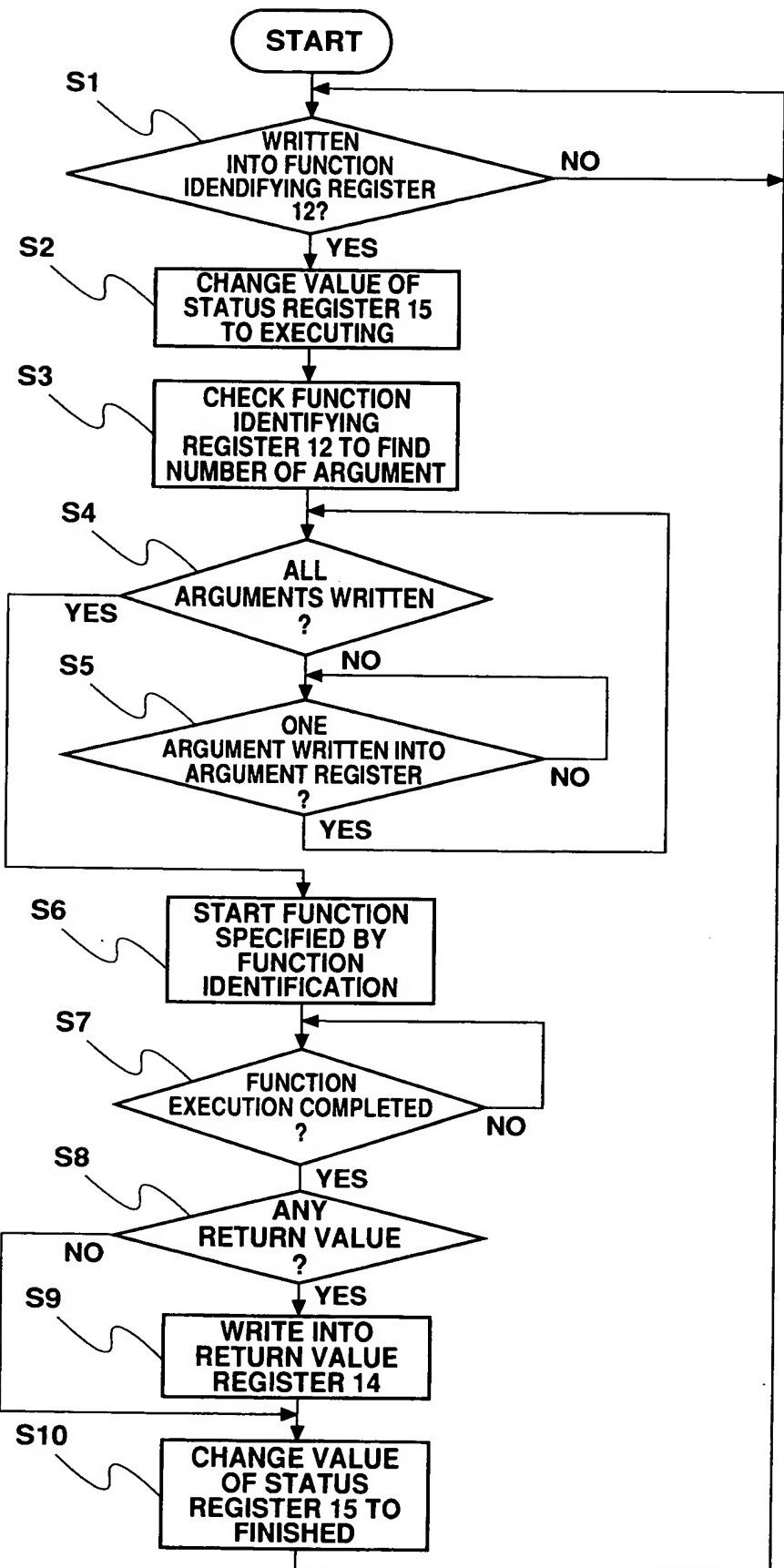
/* proto-type declaration in C-language function, corresponding to function */
void    setSeed( double seed );
double  getRandom()
```

**Fig. 4**

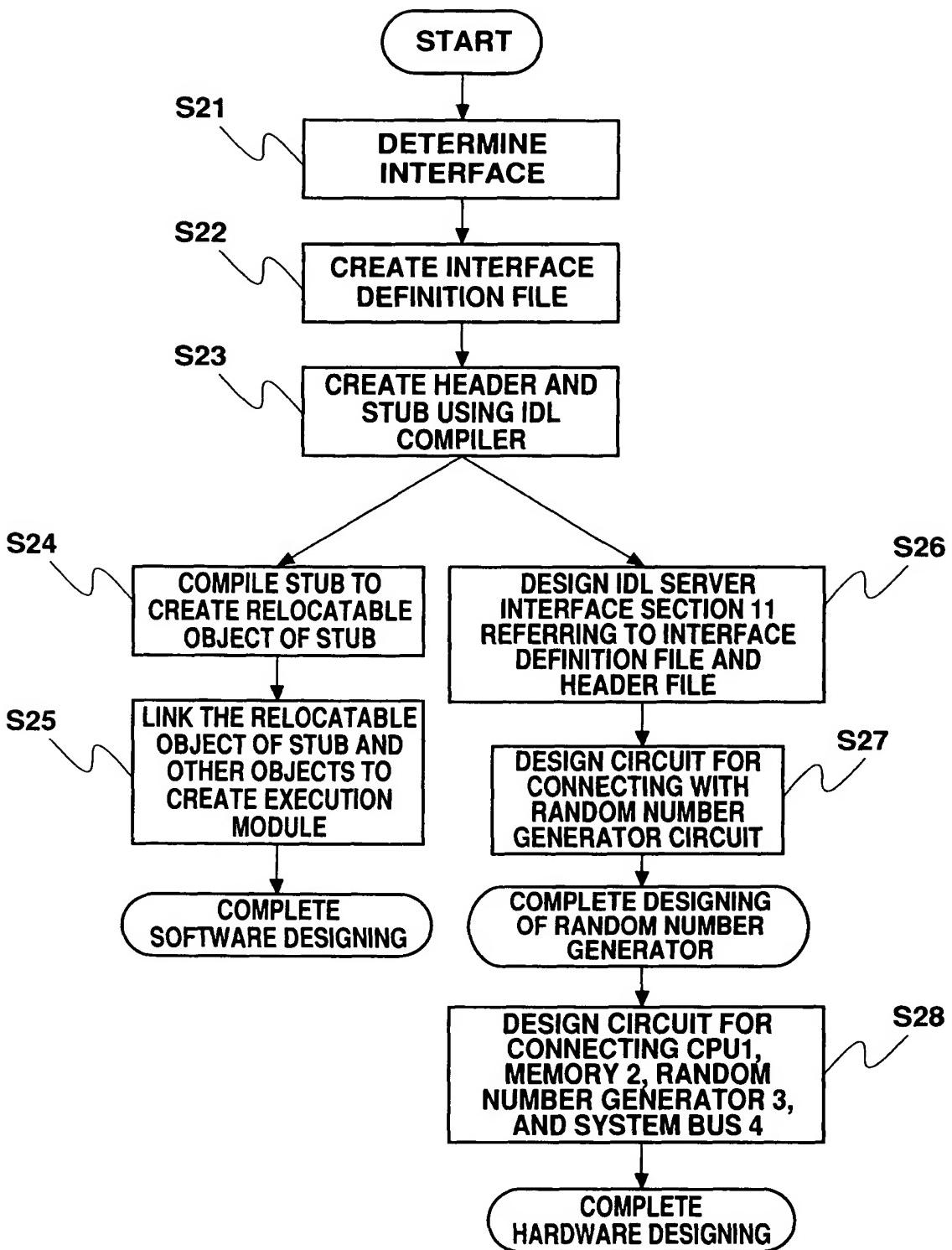
```
void setSeed( double seed )
{
    putFunctionID( FID_setSeed );
    putDoubleArg( seed );
}

double getRandom( )
{
    putFunctionID( FID_getRandom );
    while( getStatus() != Finished )
        ;
    return getDoubleReturn();
}
```

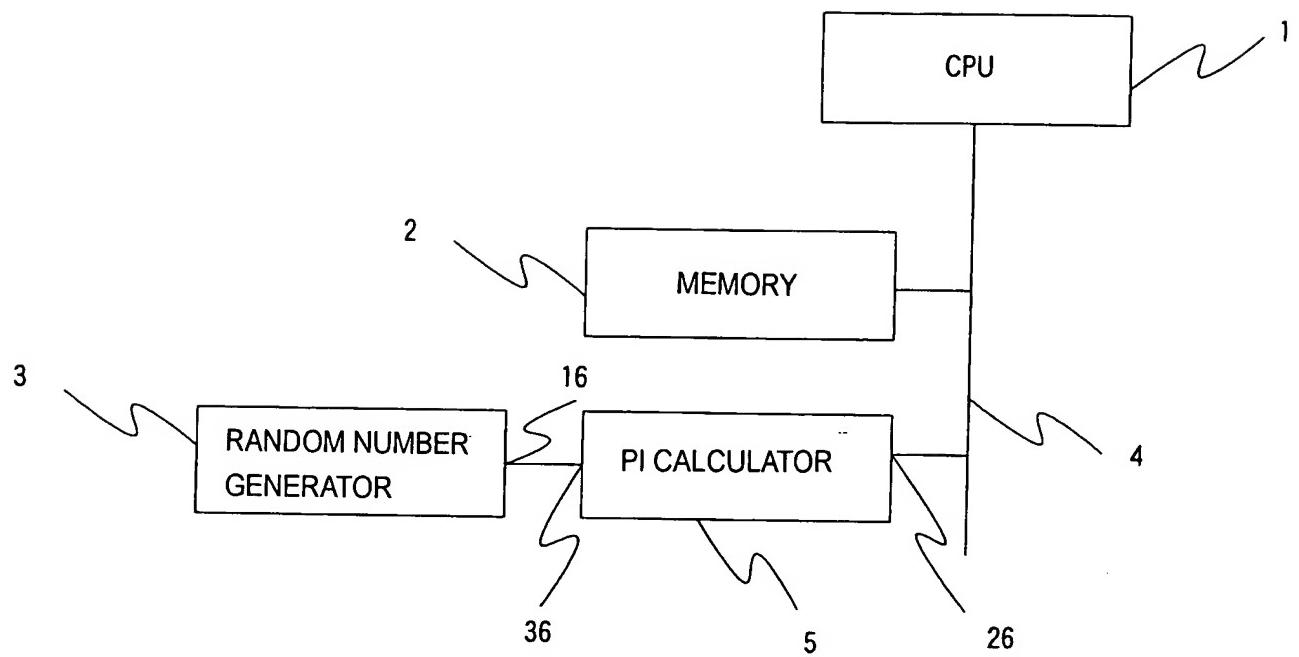
**Fig. 5**



**Fig. 6**



**Fig. 7**



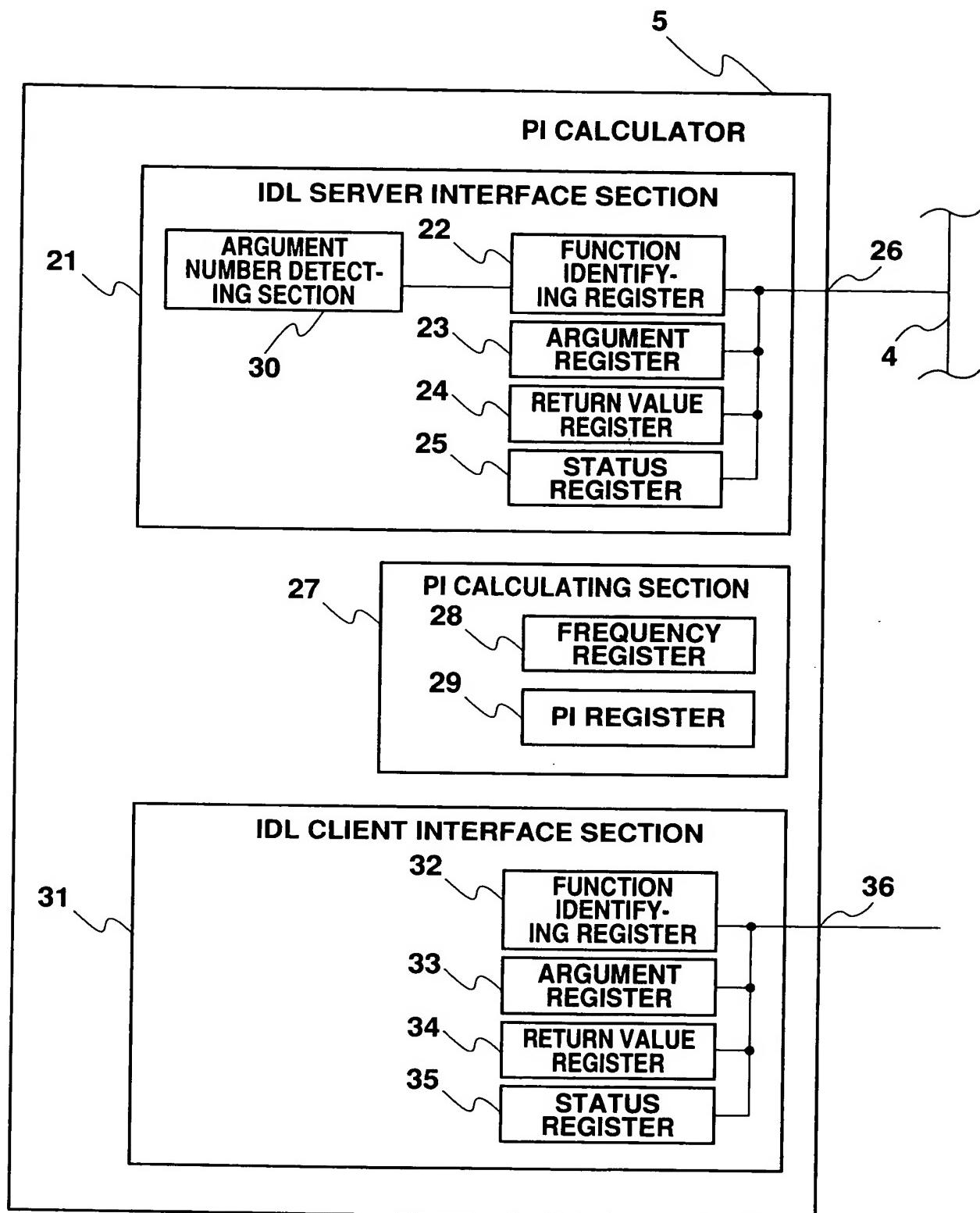
**Fig. 8**

```

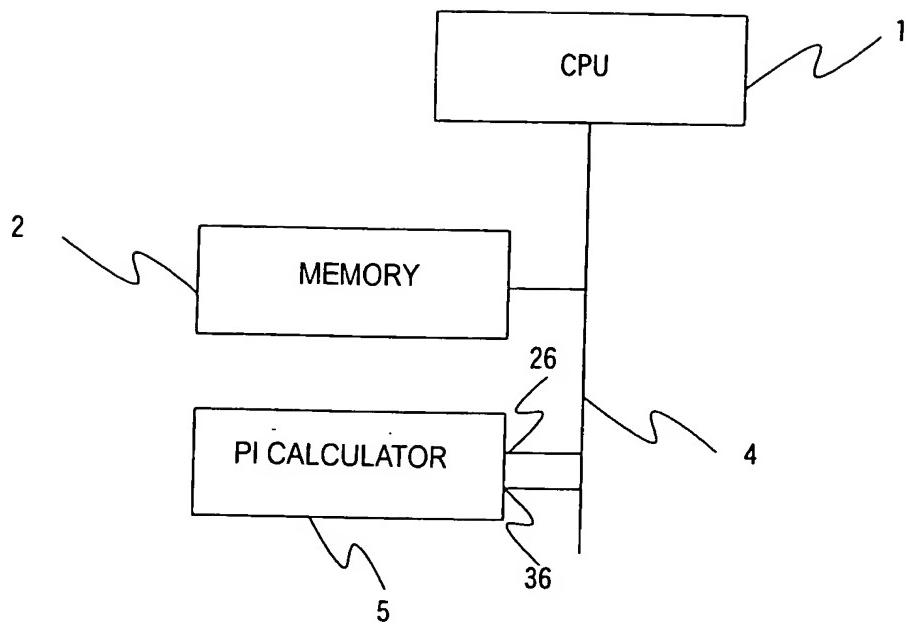
interface piCalculator{
    void setCount( in int count );
    double getPi();
};

```

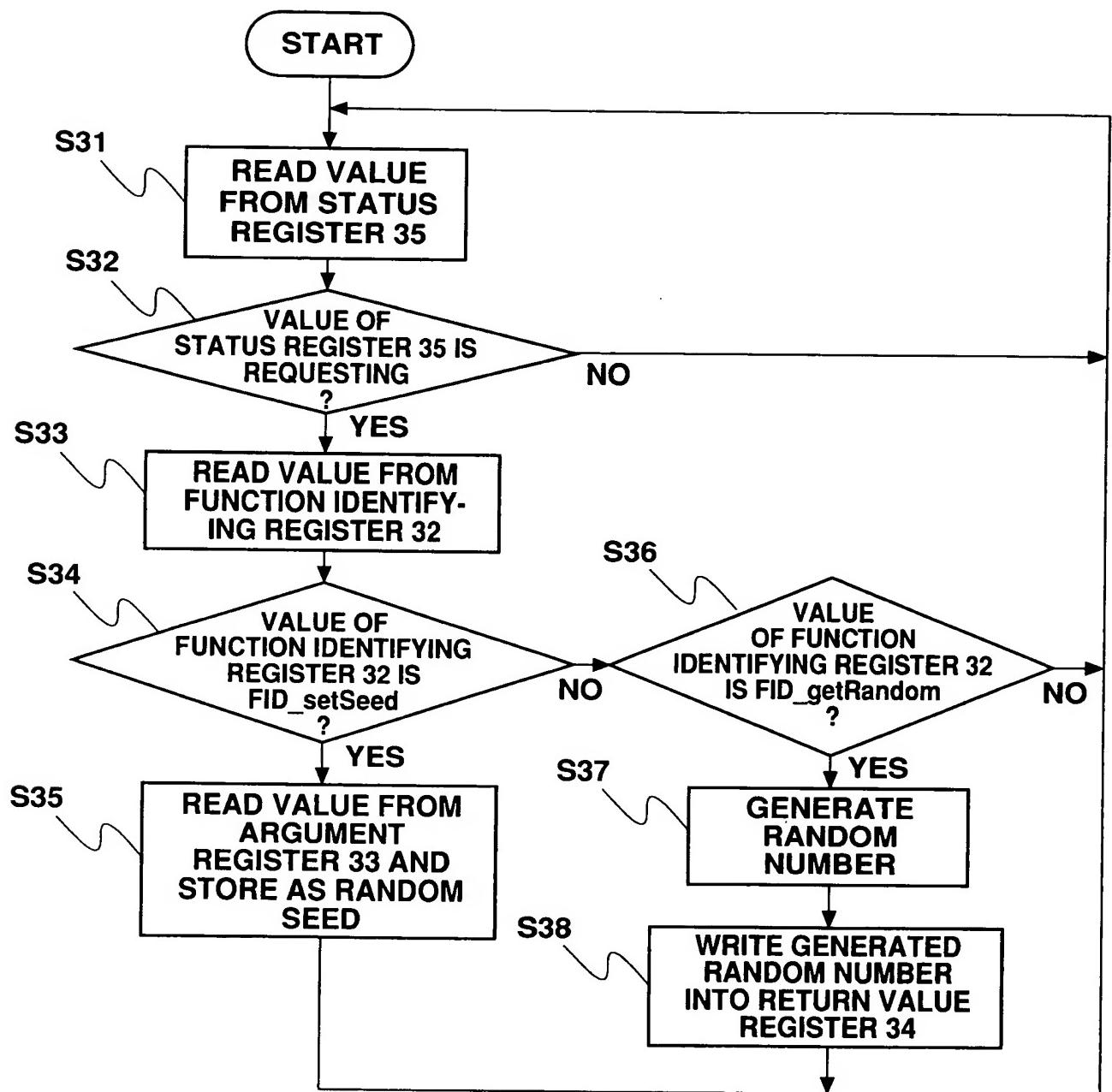
**Fig. 9**



**Fig. 10**



**Fig. 11**



**Fig. 12**

```

/* part common to IDL interface */
/* status register value */
#define Waiting      0
#define Executing   1
#define Finished    2
#define Requesting  3

/* function for accessing register */
FID    getFunctionID();
double getDoubleArg();
double putDoubleReturn( double );
int    getStatus();

/* randomGenerator unique part */
/* definition of function identification value */
#define FID_setSeed    1
#define FID_getRandom  2

/* proto-type declaration in C-language function, corresponding to function */
void    setSeed( double seed );
double  getRandom()

```

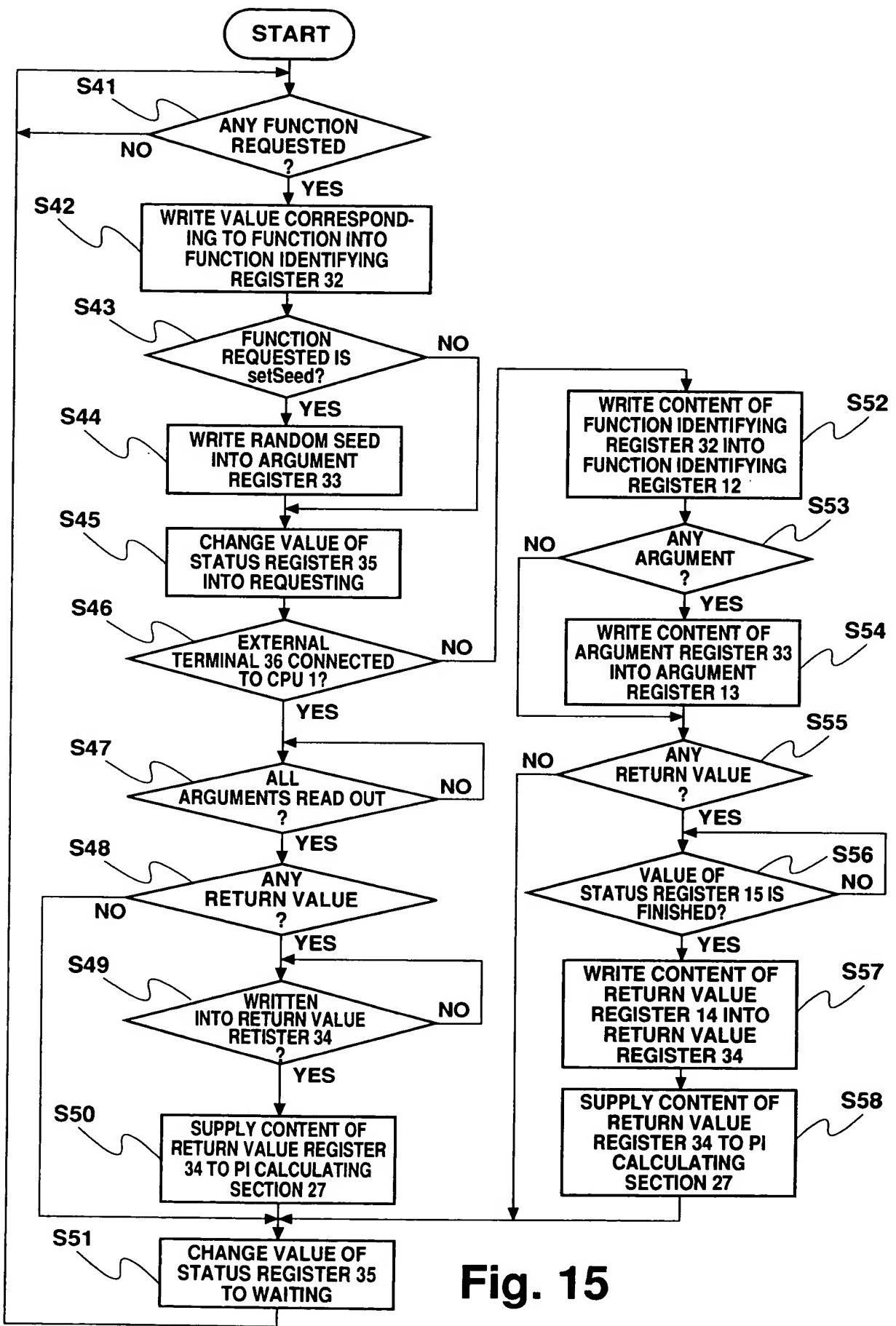
## Fig. 13

```

main()
{
    while(1){
        while( getStatus() != Requesting )
            ;
        switch( getFunctionID() ){
            case FID_setSeed:
                setSeed( getDoubleArg() );
                break;
            case FID_getRandom:
                putDoubleReturn( getRandom() );
                break;
        }
    }
}

```

## Fig. 14



**Fig. 15**